UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/735,715	12/12/2000	Jacob Dreyband	033144-004	5590

23562 7590 10/31/2003

BAKER & MCKENZIE
PATENT DEPARTMENT
2001 ROSS AVENUE
SUITE 2300
DALLAS, TX 75201

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2124	6

DATE MAILED: 10/31/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/735,715	DREYBAND ET AL.	
	Examiner	Art Unit	
	Tuan A Vu	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 December 2000* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☐ Claim(s) _____ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-71* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *12 December 2000* is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. This action is responsive to the application filed December 12, 2000.

Claims 1-71 have been submitted for examination.

Specification

2. The disclosure is objected to because of the following informalities: there appears to be a misspelled term as in "form" (pg. 4, line 14; pg. 10, line 5), and this should be "from".

Appropriate correction is required.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-7, 9-11, 13-33, 35-37, and 39-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bowman-Amuah, USPN: 6,477,580 (hereinafter Bowman), and in view of White et al., USPN: 6,438,559, (hereinafter White).

As per claim 1, Bowman discloses a method for presenting data within a computing environment including an application program interface (e.g. col. 103, line 63 to col. 105, line 6; col. 36-38 – Note: window based application implicitly discloses APIs), said method comprising the steps of:

creating a tagged data object for storing data (e.g. *software package* – col. 105, line 42-66; *bundle, message* – Fig. 185-187);

encapsulating a data element into a tagged data object to provide a tagged data including a data element (e.g. *encapsulation* – col. 299, line 1 to col. 300, line 14; Fig. 184-185);

packing the tagged data by converting the tagged data into a binary representation of tagged data (e.g. Fig. 20-22; *stream out business object* – col. 281, line 17 to col. 282, line 29; Fig. 165; data *stream* -Figs. 184-191).

But Bowman does not explicitly specify including in the encapsulated tagged data within the tagged data object a corresponding tag id and data element. Bowman, however, teaches attributes descriptors in the meta-data section comprising a header section of the object-based stream, which suggests encapsulating identification information of the data element of the stream. Further, the inclusion of identifier to associate the data bundle or packet sent over a network communication link with its content was a well-known concept in the art at the time the invention was made. White, in a method to serialize objects for distribution over a communication network environment using descriptors in serialization of class objects analogous to the object streaming and meta-data by Bowman, discloses the tagging of object content being serialized with an identifier or value for packing and deserializing (e.g. *ACI* - col. 4, lines 16-58; col. 10, line 36 to col. 11, line 9). It would have been obvious for one of ordinary skill in the art at the time the invention was made to use the tagging technique associating an identifier with the tagged content as taught by White and apply it to the stream metadata by Bowman, in case Bowman's metadata does not include such tag identification already, because this tag ID would facilitate the differentiation between data being packed and enable data handling/re-processing as well as unpacking or modification of elements packed in the message or bundle.

As per claim 2, Bowman discloses the packing of tagged data being a simple object and a complex object, and list object (e.g. col. 124, line 14 to col. 127, line 39 – Note: the use of Java or C++ based components implicitly discloses basic class, compound classes, or structure/enumeration of basic classes and compound classes objects).

As per claim 3, Bowman discloses packing a simple object by retrieving data attributes for length of an object source identifier, object size, type, value; allocating of packed memory location for object identifier length (e.g. col. 235, line 47 to col. 237, line 32); copying the object size, type, and value into the packed memory location (Note: this is inherent to the above cited portions); retrieving and copying head value and exit value into the packed memory location (e.g. *START INDEX, WS-INDEX, STREAM-END* - col. 237, line 35 to col. 238, line 66).

As per claims 4 and 5, Bowman does not explicitly specify the steps of retrieving, writing, and allocating/writing for the complex object as has been disclosed for the simple object from claim 3, but in view of the packaging of data in the retrieval of business-related complex object (e.g. col. 204, line 40 to col. 207, line 59), the limitations as recited are herein implicitly in view of the inherent presence of simple object within complex object or list objects.

As per claim 6 and 7, Bowman does not specify packing list object with retrieving of object source identifier, allocating memory in a packed memory location to accommodate the list object source identifier length; retrieving and copying list head value and list exit value into the packed memory location; but in view of the rationale used in addressing claims 4 and 5, these limitations are also implicitly disclosed because of the inherent presence of simple objects and complex objects in structure or enumeration, i.e. list, object so well-known in object-oriented language.

Further, Bowman does not explicitly disclose retrieving list array object and copying it to the packed memory location. But, in view of the inherent array structure in structure or enumeration of simple and complex objects in C++ or Java, this limitation is also implicitly disclosed as per the same rationale used for claims 4 and 5.

As per claim 9, see Bowman (e.g. Fig. 20-22; *stream out business object* – col. 281, line 17 to col. 282, line 29; Fig. 165; data *stream* -Figs. 184-191).

As per claim 10, refer to claim 2.

As per claim 11, Bowman discloses data wrapping (e.g. *Wrapper component* - Fig. 81).

As per claim 13, Bowman discloses Java and C++ constructs which inherently include list or enumeration of objects of simple and complex type (see claim 2).

As per claim 14, this claim includes the encapsulation of data type, tag id, and writing thereof to the tagged data object and these limitations have been addressed in claim 3 and 4.

As per claim 15, Bowman discloses the use of Java objects, hence has implicitly disclosed one of the following data type: integer, float, byte, char string, a java object, a null data, a primitive type, a compound type, and a list type.

As per claim 16, Bowman (in combination with White) discloses tag identifier with of type integer (see White from claim 1).

As per claim 17, Bowman with White's teachings discloses serializing of tagged data and compacting it in a stream for transmission, hence has implicitly disclosed a tagging process following a linear sequence, i.e. sequential tagging with determining a sequence.

As per claim 18, Bowman with White's teachings discloses including a data, a position and a tag element (refer to claim 3; Fig. 109 – Note: Index position use in writing data by

Bowman discloses including an position and packet layout inherently encompasses boundaries position of data compacted in packet).

As per claim 19, Bowman does not explicitly specify converting of first type of tagged data to second type of data for a change in properties; but the concept for converting the order of data type (e.g. network-bound integer converted into local host-based integer and vice-versa, as per Java/C++ *ntohs* or *htons* functions) for allowing data type to be communicated through the internet medium was a well-known concept at the time the invention was made. Hence, Bowman's disclosed communication of Java or C++ objects implicitly discloses such conversion to provide for a communication properties adjustment or change as claimed.

As per claims 20 and 21, by virtue of the rejections of claim 2 and claim 19 above, the limitations of these claims are implicitly disclosed.

As per claim 22, Bowman discloses a method for presenting data within a computing environment including an application program interface comprising the steps of unpacking tagged data from a binary representation; creating tagged data object for storing the tagged data; and extracting a data element for the tagged data (e.g. Fig. 185, 187; *unpackaged* - col. 300, line 39 to col. 301, line 29-- Note: in view of the teachings on packing data into package or stream to be sent in packet over the internet by Bowman as mentioned in claim 1, the steps of unpacking, creating a storage for the unpacked data received over the internet, and the extracting of object being tagged are implicitly disclosed).

As per claim 23, this corresponds to claim 2, in view of the above rejection, is rejected using claim 2 and claim 22 rationale.

As per claim 24, Bowman does not specify the steps of retrieving the simple head value and simple exit value; allocating memory in an unpacked memory and copying of simple object size, type and value into said unpacked memory. Official notice is taken that subjecting packets received from the internet into a host or routing, or a gate way machines to unpacking and buffer storage was a well-known concept in the art at the time the invention was made. In view of the teachings for unpackaging of data by Bowman above and the well-known unpacking of data, it would have been obvious for one skill in the art at the time the invention was made to provide the unpacking of the tagged data as taught by Bowman/White using the well-known technique of unpacking/storage above because this would enable correct extraction of data based on boundaries locations and allocation of correct memory resources.

As per claims 25 and 26, the limitations as to unpack a complex object would also have been obvious by virtue of the inherency of simple object in a complex objects as mentioned in claims 4-5 and the rejection used in claim 24 above.

As per claims 27 and 28, the rationale used for claims 6-7 and 25-26 is herein applied.

As per claim 29, Bowman does not explicitly specify determining a tag sequences but by virtue of the rejection of claim 17 for packing, the limitation would also have been obvious because the sequential unpacking of data being associated with a tag sequence would provide accurate mapping of data thus unraveled according to a predetermined packed sequences, hence provide correctness of serialized data received for utilization.

As per claim 30, Bowman does not specify extracting data with determining the type to provide the tag id; and writing the data element into the tagged data object. But in view of White's teaching to provide a tagging associated with an identifier in order to facilitate the

reprocessing of data manipulated at the receiving end, it would have been obvious for one skill in the art at the time the invention was made to provide the determining of type and writing of tagged data based on the tag id as suggested by White to Bowman's unpacking process because this would help impart the correct memory storage to the corresponding data having a specific length identified by the tag Id.

As per claims 31 and 32, see rejection of claims 15 and 16 respectively.

As per claim 33, in view of the unpacking as taught by Bowman and the rationale used in claims 22-24 and in claim 18 above, this limitation would also have been obvious by virtue of the adding of element in the tagged data as mentioned in the above rejection.

As per claims 35-37, refer to the rationale of claims 9-11, respectively.

As per claims 39-42, refer to the rationale used in claims 7, 19, 20, and 21, respectively.

5. Claims 8, 12, 34, 38, and 43-71 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bowman-Amuah, USPN: 6,477,580, and White et al., USPN: 6,438,559, as applied to claims 1, 10, 22, 36 (for claims 8, 12, 34,38) and further in view of Miloushev et al., USPN: 6,226,692 (hereinafter Miloushev).

As per claim 8, Bowman does not explicitly disclose tagged data object being an universal data container that is platform and architecture independent; but in view of the use of Java byte code transmission (e.g. col. 99, lines 7-40), the transmission of tagged object being in a architecture and platform independent form is disclosed. Further, Bowman discloses tagged object data for providing broad access to manipulation and aggregation of structured data and unstructured data (e.g. *view configurator, maximum maintainability and extensibility* - col. 248, line 28 to col. 259, line 45; *LUW* - Fig. 108-129, 163-191 – Note: context retrieving and

selecting appropriate objects from requests is equivalent to broad access for data manipulation and aggregation, structured data are metadata or compiled Java code and unstructured data are raw uncompiled data).

But Bowman fails to specify that the universal data container is language independent. Miloushev, in a method to construct software components from parts and extending them into models, analogous to modeling via Case Tools to enhance scalability and reusability by Bowman (Bowman: col. 131-132; col. 174, line 33 to col. 175, line 24; Fig. 50-51), discloses modeling using COM and Case Tools with emphasis on generating language-independent constructs to enable larger range of programming language applicability of execution environments (e.g. col. 17, line 39-52). It would have been obvious for one of ordinary skill in the art at the time the invention was made to use language-independent modeling technique associating as taught by Miloushev and apply it to the model of objects packed in streamed metadata by Bowman, in case Bowman's metadata does not include such language-independent model constructs already, because this would help extend the applicability of the software to build and distribute to the requesting clients operating from any executing platform environment.

As per claim 12, Bowman does not specify including of named tree with a field name connected with a value. But in view of the including of language-independent model (Note: modeling is equivalent to tree structure implementation) in the stream as mentioned in claim 8 above, this limitation would have obvious for the same rationale as used in claim 8 and also the association of tree with field name as metadata would enhance the utilization and re-processing of data tagged and stored in the package.

As per claim 34, this is a unpacking version of claim 8, hence is rejected using the rationale as set forth therein and the motivation to provide the unpacking of data in claim 22.

As per claim 38, this is a unpacking version of claim 12, hence is rejected using the rationale as set forth therein and the motivation to provide the unpacking of data in claim 22.

As per claim 43, Bowman discloses a method for presenting data within a computing environment including an application program interface prescribed for data conversion and wire formatting specification (e.g. col. 103, line 63 to col. 105, line 6; col. 36-38 – Note: window based application implicitly discloses APIs), said method comprising the steps of:

creating a tagged data object (e.g. *software package* – col. 105, line 42-66; *bundle, message* – Fig. 185-187); wherein the tagged data object comprises a universal data container that is platform and hardware independent (e.g. col. 99, lines 7-40 – Note: Java platform independency is implicitly disclosed); said tagged data object providing broad access to manipulation and aggregation of structured data and unstructured data (e.g. *view configurator, maximum maintainability and extensibility* - col. 248, line 28 to col. 259, line 45; *LUW* - Fig. 108-129, 163-191 – Note: context retrieving and selecting appropriate objects from requests is equivalent to broad access for data manipulation and aggregation);

encapsulating a data element into a tagged data object to provide a tagged data including a data element (e.g. *encapsulation* – col. 299, line 1 to col. 300, line 14; Fig. 184-185);

packing the tagged data by converting the tagged data into a binary representation of tagged data (e.g. Fig. 20-22; *stream out business object* – col. 281, line 17 to col. 282, line 29; Fig. 165; data *stream* -Figs. 184-191);

transmitting the tagged data transmission (e.g. Fig. 105-107);

unpacking the tagged data from a binary representation; creating tagged data object for storing the tagged data; and extracting a data element for the tagged data (e.g. Fig. 185, 187; *unpackaged* - col. 300, line 39 to col. 301, line 29- Note: re claim 22).

But Bowman does not specify that the universal data container is language independent. But this limitation has been addressed in claim 8 above using Miloushev. Nor does Bowman explicitly specify including in the encapsulated tagged data within the tagged data object a corresponding tag id; but this also has been addressed in claim 1 above using White.

As per claims 44-49, these claims correspond to claims 2-7 respectively; hence are rejected likewise, respectively.

As per claims 50-55, these claims correspond to claims 23-28 respectively; hence are rejected likewise, respectively.

As per claims 56-59, refer to rejections of claims 36-39 respectively.

As per claim 60, this claim corresponds to claim 14, hence is rejected using the same rationale as set forth therein.

As per claims 61-67, refer to corresponding rejections of claims 15-21 respectively.

As per claims 68-71, refer to corresponding rejections of claims 30-33 respectively.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Pub No 2002/0073080 to Lipkin, disclosing metadata and tagged Java application with COM/ORB/EJB.

U.S. Pat No. 5,907,704 to Gudmundson et al., disclosing reuseable containers framework and hierarchies of elements.

U.S. Pat No. 6,292,933 to Bahrs et al., disclosing replacing class identifier code for deserializing class objects.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 872-9306 (for formal communications intended for entry)

or: (703) 746-8734 (for informal or draft communications, please label "PROPOSED" or "DRAFT")

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

VAT
October 23, 2003

TODD INGBERG
PRIMARY EXAMINER